Semester Project: **Searching Interactivity Deep Inside Some Feature Space**

Mehmet Sahin, 2097921

Abstract: Artificial neural networks are pretty good at finding things they've been taught. Today, we can say that there are many classifications made in this way. Thanks to transfer learning, these neural networks can be developed to classify inputs. As we all know, we can divide the neural networks used for discrimination into two main parts. These are the convolution part and the classification part. In this project, we will be able to distinguish between new and old things for neural networks as well as classification.

I. Introduction

In this project, our goal is to make an intelligent machine that recognizes food in an animal shelter. We can assume that different leftovers are constantly coming into this animal shelter. For this reason, we need a machine that can separate newly arrived dishes from the old ones and learn new dishes over time. I think the communication of smart machines with animals will develop before humans. For this reason, the specular part of the project will be quite a developer.

A. Implementation part

Some simplifications have been made to minimize the difficulties of this scenario. The background color is set to white to facilitate image processing. Due to the difficulties of working with animal foods, food models similar to small-grain structures have been created. According to our scenario, foods should be distinguished. New foods should be noticed. In order to reduce the processing load and create a large model, new foods must become part of the model over time. Therefore, a model that constantly improves itself and increases the number of data is needed.

As part of this project, our model should be able to distinguish between new and old things.

B. Speculation part

We will explain how the part we are implementing can be used for interaction. In the part where we implemented it, we provided the necessary background for intelligent machines to interact with their environment. In our scenario, we distinguish between meals coming to an animal shelter. They probably come from waste meals and very different dishes. Our goal here is to distribute the food in a way that makes the animals the happiest.

With the network to be trained for each animal, the animals' favorite food can be found. First, let's assume that all future dishes are correctly classified. Inputs and outputs are needed for our neural network to develop itself. Here, there is a need to evaluate how much animals love their food. For this, it can be observed how much the animals eat the food. With this data, a Fully Connected Network can be created. We can assume that the input of this network is the feature vector of the meal and the output is the liking rate for the food. Thus, a network can be obtained that estimates the likelihood of food for each animal. Then, by calculating the collective happiness in this animal shelter, the best sharing of meals can now be achieved.

II. Solution Approach and Results

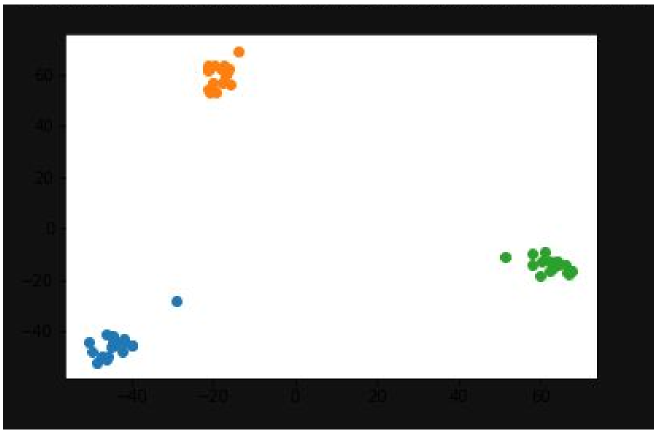
We can examine the important methods used in the project one by one.

A. Image Processing

We use real time images in the project. A 1920x1080 pixel camera was preferred to ensure sufficient success in the post-image processing stages. Images can be easily processed thanks to the white background. Due to the small-grain assumption of the foods, many connection problems may occur in images. To prevent these, "dilation method" can be used. After separating the animal foods as binary, they can be separated from the images with the help of "Connected Component Labeling". The simple appearance of the image greatly increases the success of networks.

B. Feature Extraction

To be able to cluster inputs, feature vectors must be omitted. The distribution of feature vectors used for many general purposes did not help separate the sets. To solve these problems, a convolution neural network focused on a single goal was decided. In this project, an expert network was used to separate substances called bit / expert / substance. This network has been trained by Google.



After this stage, we can say that the basic clusters that could not be separated before are separated quite well.

C. Principal Component Analysis

Feature vectors are very long vectors. Clustering algorithms have to reduce the size of these vectors because of the operational costs. Principal component analysis was used to reduce the size in this project. In the project, the feature vector length was reduced from 2048 to 10. We can say that 98 percent of the PCA variance is covered. Here, we can say that the reduced dimension corresponds to the real vector quite well.

D. K-Means Clustering

In this project, we first start with 3 basic clusters. Kmeans is a clustering algorithm related to the distance of clusters to the center. Specially for this project, the elements around the cluster centers are analyzed. Thus, we determine the regions around the cluster centers as belonging to that cluster. We assume that the regions around cluster centers are only in that cluster. In addition, the feature vectors must be sufficiently separated for our algorithm to work well enough.

We can analyze the locations of newly arrived feature vectors. If the new feature vectors are within the cluster regions we just mentioned, it can be said that this feature vector is an element of this set. On the other hand, if this new feature vector is not in this cluster region, a new set can be created for this feature vector.

III. Presentation and GitHub

Our presentation is recorded at the Zoom Meeting. In addition, a demo of test setup will be online in the GitHub repository.

All sources of this project will be online GitHub repository.

Github: https://github.com/mehmetsahin1001/ME536Project/tree/master

IV. Summary

The algorithm used to separate new and old is explained in the project. For this algorithm to work well enough, feature vectors must be extracting sufficiently successfully. Thanks to the good feature vector model in our project, our system can operate with high performance. The code in the GitHub repository can be modified and used in other projects.

References

Manjunath Jogin(2018), Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning

Benyamin Ghojogh(2019), Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review

Sourish Dey(2018), CNN application on structured data-Automated Feature Extraction

Krut Patel(2020), Image Feature Extraction: Traditional and Deep Learning Techniques